IN THE CLAIMS:

The following is a complete listing of the claims, reflects all changes proposed to be made thereto, and replaces all earlier listings and versions:

1. (currently amended): A method of generating an encoded representation of a markup language document comprising syntactic elements, said method comprising the steps of:

parsing the markup language document to identify at least one syntactic element of that document;

identifying a type of the element;

processing the element by applying a hash function thereto ~~if the type is a first type~~, the hash function generating a numeric code from the element; and

generating the encoded representation including the numeric code;

wherein first and second syntactic elements respectively comprise a start tag and an end tag, being a first pair of tags, and said processing of said start tag and of said end tag generates corresponding hashed start and end tags;

wherein corresponding hashed start and end tags for the first pair of tags are incorporated into the encoded representation of the document;

wherein the document further includes a second pair of tags comprising respective start and end tags, the second pair of tags being nested within the first pair of tags in the document, and

said method comprising further steps of:

processing the second pair of tags to form corresponding second hashed start and end tags; and

augmenting the encoded representation of the document using the corresponding second hashed start and end tags so that the second hashed start and end tags indicate a nesting in relation to the hashed start and end tags for the first pair of tags which is equivalent to the nesting of the second pair of tags within the first pair of tags.

2. (previously presented): A method according to claim 1, wherein the parsing is event-based parsing.

3. (previously presented): A method according to claim 1, wherein the numeric code is determined using one of:

a hash algorithm;

a first reference to the hash algorithm dependent upon an associated Universal Reference Indicator;

a second reference to the hash algorithm dependent upon an associated namespace; and

a third reference to the hash algorithm dependent upon an associated Extended Markup Language declaration.

4. (previously presented): A method according to claim 1, wherein the element type is one of:

one of a structural element and a part thereof;

a definition of the structural element;

a declaration of the structural element; and

a match for the structural element.

5. (previously presented): A method according to claim 4, wherein the structural element is a tag.

6. (previously presented): A method according to claim 1, wherein the numeric code is a unique code for the one syntactic element, the element having less than a first number of characters.

7. (previously presented): A method according to claim 1, wherein the hash function is not guaranteed to produce a unique numeric code for the one syntactic element, in situations where the element is constrained, to a probability level, in terms of at least one of (i) a number of characters in the element and (ii) a permissible number of permutations of characters in the element.

8. and 9. (canceled).

10. (previously presented): A method according to claim 1 comprising, for another one of the syntactic elements, the further steps of:

identifying a type of the other element;

processing the other element by applying the hash function thereto, thereby to generate a second numeric code from the other element; and

augmenting the enclosed representation of the markup language document using the second numeric code,

wherein said processing of the one element and said processing of the other element ensure that if a first relationship exists between the one element and the other element, then a second relationship which is representative of the first relationship, exists between the numeric code of the one element and the second numeric code of the other element.

11. (previously presented): A method according to claim 10, wherein:

the one element is a start tag;

the other element is an end tag;

the numeric code of the one element is a corresponding hashed start tag, and

the second numeric code of the other element is a corresponding hashed end tag.

12. (previously presented): A method according to claim 11, wherein:

the end tag is a first modification of the start tag; and

the hashed end tag is a second modification of the hashed start tag, the second modification being representative of the first modification.

13. (original) A method according to claim 12, wherein:

the end tag is the same as the start tag apart from having a distinguishing character incorporated therein; and

the hashed end tag is at least one of:

the same as the hashed start tag;

the same as the hashed start tag apart from having a distinguishing character incorporated therein; and

the hashed start tag having been processed by an operator.

14. (canceled).

15. (previously presented): A method according to claim 1 comprising, prior to said augmenting step, a further step of:

concatenating the first hashed start tag with the second hashed start tag, and concatenating the first hashed end tag with the second hashed end tag, to thereby form respective extended hashed start and end tags for said second pair,

wherein said augmenting step is performed using the respective extended hashed start and end tags for the second pair, and

the extended hashed start and end tags indicate a nesting in relation to the hashed start and end tags for the first pair of tags which is equivalent to the nesting of the second pair of tags within the first pair of tags.

16. (previously presented): A method according to claim 20, wherein said well-formedness checking step comprising checking the markup language document against the syntactic rule by numerically comparing corresponding numeric code of elements in the encoded representation of the markup language document.

17. (previously presented): A method according to claim 16, wherein said numerically comparing step is succeeded by a further step of:

string-comparing, in accordance with the syntactic rule, corresponding non-processed representations of non-tag elements.

18. and 19. (canceled).

20. (previously presented): A method according to claim 1, comprising a further step of:

checking the well-formedness of the encoded representation of the document against a syntactic rule.

21. (previously presented): A method according to claim 16, wherein the syntactic rule relates to proper nesting of tags and said checking step comprises sub-steps of:

performing a numerical comparison across hashed tags in the encoded representation of the document, thereby to identify the first hashed start and end tags and the second hashed start and end tags; and

verifying that the second hashed start and end tags indicate a proper nesting in relation to the first hashed start and end tags.

22. (previously presented): A method according to claim 21, wherein the numerical comparison is followed by a further step of:

performing a string comparison, in accordance with the syntactic rule, across non-processed parts of respective tags in the encoded representation of the document.

23. to 25. (canceled).

26. (previously presented): A method according to claim 16, wherein said well-formedness checking step is one of (a) succeeded by, (b) included in, and (c) replaced by a validation step against a validation reference document VRD, said validation step comprising sub-steps of:

(a) processing the VRD, said processing comprising, for a syntactic element in the VRD, sub-sub-steps of:

(i) identifying a type of the syntactic element of the VRD; and

(ii) processing the syntactic element by applying a hash function thereto if the type is the first type, the hash function generating a numeric code from the element; and

(b) checking the encoded representation of the markup language document against the processed VRD, said checking comprising a sub-sub-step of numerically comparing corresponding numeric codes of the elements.

27. to 33. (canceled).

34. (previously presented): An apparatus for generating an encoded representation of a markup language document comprising syntactic elements, said apparatus comprising:

parsing means for parsing the markup language document to identify at least one syntactic element of that document;

identifying means for identifying a type of the element;

processing means for processing the element by applying a hash function thereto, said hash function generating a numeric code from the element; and

generating means for generating the encoded representation including the numeric code,

wherein first and second syntactic elements respectively comprise a start tag and an end tag, being a first pair of tags and said processing of said start tag and of said end tag generates corresponding hashed start and end tags;

wherein corresponding hashed start and end tags for the first pair of tags are incorporated into the encoded representation of the document;

wherein the document further includes a second pair of tags comprising respective start and end tags, the second pair of tags being nested within the first pair of tags in the document;

wherein said processing means processes the second pair of tags to form corresponding second hashed start and end tags; and

wherein said generating means augments the encoded representation of the document using the corresponding second hashed start and end tags so that the second hashed start and end tags indicate a nesting in relation to the hashed start and end tags for the first pair of tags which is equivalent to the nesting of the second pair of tags within the first pair of tags.

35. to 38. (canceled).

39. (previously presented): A computer-executable program which is stored on a computer-readable storage medium and which is configured to make a computer execute a procedure to generate an encoded representation of a markup language document comprising syntactic elements, said program comprising:

code for parsing the markup language document to identify at least one syntactic element of that document;

code for identifying a type of the identified element;

code for processing the identified element by applying a hash function thereto, the hash function generating a numeric code from the identified element; and

code for generating the encoded representation including the numeric code,

wherein first and second syntactic elements respectively comprise a start tag and an end tag, being a first pair of tags, and said code for processing of said start tag and of said end tag generates corresponding hashed start and end tags;

wherein corresponding hashed start and end tags for the first pair of tags are incorporated into the encoded representation of the document;

wherein the document further includes a second pair of tags comprising respective start and end tags, the second pair of tags being nested within the first pair of tags in the document;

wherein said code for processing processes the second pair of tags to form corresponding second hashed start and end tags; and

wherein said code for generating augments the encoded representation of the document using the corresponding second hashed start and end tags so that the second hashed start and end tags indicate a nesting in relation to the hashed start and end tags for the first pair of tags which is equivalent to the nesting of the second pair of tags within the first pair of tags.

40. to 43. (canceled).

44. (previously presented): A computer program product including a computer readable storage medium having encoded thereon a computer program which is configured to make a computer execute a procedure to generate an encoded representation of a markup language document comprising syntactic elements, said program comprising:

code for parsing the markup language document to identify at least one syntactic element of that document;

code for identifying a type of the element;

code for processing the element by applying a hash function thereto, the hash function generating a numeric code from the element; and

code for generating the encoded representation including the numeric code,

wherein first and second syntactic elements respectively comprise a start tag and an end tag, being a first pair of tags, and said code for processing of said start tag and of said end tag generates corresponding hashed start and end tags;

wherein corresponding hashed start and end tags for the first pair of tags are incorporated into the encoded representation of the document;

wherein the document further includes a second pair of tags comprising respective start and end tags, the second pair of tags being nested within the first pair of tags in the document;

wherein said code for processing processes the second pair of tags to form corresponding second hashed start and end tags; and

wherein said code for generating augments the encoded representation of the document using the corresponding second hashed start and end tags so that the second hashed start and end tags indicate a nesting in relation to the hashed start and end tags for the first pair of tags which is equivalent to the nesting of the second pair of tags within the first pair of tags.

45. and 46. (canceled).

47. (previously presented): An encoded representation of a markup language document comprising syntactic elements, the encoded representation having been produced by a method comprising:

parsing the markup language document to identify at least one the syntactic element of that document;

identifying a type of the element;

processing the element by applying a hash function thereto, the hash function generating a numeric code from the element; and

generating the encoded representation including said numeric code,

wherein first and second syntactic elements respectively comprise a start tag and an end tag, being a first pair of tags, and said processing of said start tag and of said end tag generates corresponding hashed start and end tags;

wherein corresponding hashed start and end tags for the first pair of tags are incorporated into the encoded representation of the document;

wherein the document further includes a second pair of tags comprising respective start and end tags, the second pair of tags being nested within the first pair of tags in the document, and

said method comprising further steps of:

processing the second pair of tags to form corresponding second hashed start and end tags; and

augmenting the encoded representation of the document using the corresponding second hashed start and end tags so that the second hashed start and end tags indicate a nesting in relation to the hashed start and end tags for the first pair of tags which is equivalent to the nesting of the second pair of tags within the first pair of tags.

48. (previously presented): An apparatus for generating an encoded representation of a markup language document comprising syntactic elements, said apparatus comprising:

a processor;

a memory for storing (i) the document, and (ii) a program which is configured to make the processor execute a procedure to generate the encoded representation,

wherein said program comprises:

code for parsing the markup language document to identify at least one syntactic element of that document;

code for identifying a type of the element;

code for processing the element by applying a hash function thereto, the hash function generating a numeric code from the element; and

code for generating the encoded representation including said numeric code,

wherein first and second syntactic elements respectively comprise a start tag and an end tag, being a first pair of tags, and said code for processing of said start tag and of said end tag generates corresponding hashed start and end tags;

wherein corresponding hashed start and end tags for the first pair of tags are incorporated into the encoded representation of the document;

wherein the document further includes a second pair of tags comprising respective start and end tags, the second pair of tags being nested within the first pair of tags in the document;

wherein said code for processing processes the second pair of tags to form corresponding second hashed start and end tags; and

wherein said code for generating augments the encoded representation of the document using the corresponding second hashed start and end tags so that the second hashed start and end tags indicate a nesting in relation to the hashed start and end tags for the first pair of tags which is equivalent to the nesting of the second pair of tags within the first pair of tags.

49. to 58. (canceled).

59. (previously presented): A method according to claim 1, further comprising the step of validating the markup language document against a validation reference document (VRD), said validating comprising sub-steps of:

(a) processing the markup language document, for each document tag identified therein, if the document tag is not a first document tag in a corresponding markup language document tag hierarchy, said processing comprising the sub-sub-steps of:

(i) determining a hierarchy position of the document tag;

(ii) determining an extended numeric code of the document tag concatenated with a numeric code of a previous document tag in the document tag hierarchy; and

(iii) storing the extended numeric code of the document tag if the document tag is more deeply nested than a previous document tag;

(b) processing the VRD, for each tag identified therein, if the tag is not a first tag in a corresponding tag hierarchy, said processing comprising sub-sub-steps of:

(i) determining a hierarchy position of the tag;

(ii) determining an extended numeric code of the tag concatenated with a numeric code of a previous tag in the corresponding tag hierarchy; and

(iii) storing the extended numeric code of the tag in a list; and

(c) validating the markup language document if the extended numeric code of the document tag is one of found in the list and is a valid subset of a member of the list.

60. (previously presented): A method according to claim 1, further comprising the step of determining a compressed representation of the syntactic element if the type is not a first type of element.

61. (canceled).

62. (previously presented): An apparatus according to claim 34, wherein each of said means forms a part of an embedded computer system.